

Ahmed Bilal Sahin

+90-530-099-5648 ✉ sahinahmedblal48@gmail.com [linkedin.com/ab-sahin](https://www.linkedin.com/ab-sahin) github.com/blisahn ahmedshn.com

Profile

Full-Stack Developer experienced in Java (Spring Boot) and C# (.NET Core) for backend, Next.js/React for frontend. Built event-driven microservice systems using Kafka, RabbitMQ, and patterns like Saga, CQRS, and Transactional Outbox. Passionate about designing scalable, maintainable architectures with domain-driven principles.

Experience

Simovate

Feb 2026 – May 2026

Software Engineer Intern

Ankara, Turkiye

- Refactored core library to optimize database query performance and reduce server load through async/await pattern improvements, significantly decreasing response times across critical endpoints.
- Implemented a Permission-Based Access Control (PBAC) library with bitwise flag permission checks and a two-layer security model spanning Next.js frontend and .NET Core backend.
- Built an automated component discovery system using ts-morph to scan and register protected UI components (ProtectedDiv, ProtectedButton, ProtectedTable) into a build-time manifest, eliminating manual permission mapping.

Vakıfbank

Aug 2025 – Sep 2025

Software Engineer Intern

Istanbul, Turkiye

- Built a .NET Core Web API applying the Clean Architecture patterns and Onion architecture standards used in the enterprise systems.
- Implemented JWT authentication, dependency injection, and Entity Framework Core following security protocols observed in the corporate codebase.
- Gained hands-on experience about enterprise banking infrastructure, Agile workflows, and code review practices.

Projects

Fraud Detection - In Development [🔗](#) | Java 21, Spring Boot, PostgreSQL, Apache Kafka, Outbox-Saga Pattern, Microservice

- Decomposed a monolithic fraud detection system into two independently deployable microservices (Transaction Service, Fraud Analysis Service) communicating over Kafka.
- Built idempotent Kafka consumers with exactly-once processing semantics, ensuring reliable event-driven communication between services.
- Designed a choreography based Saga with compensating actions and timeout based fallbacks, achieving exactly-once processing through idempotent consumers.

V - Stock [🔗](#) | .Net Core, Angular, Tailwind CSS, Redis, Docker, EF Core, SQL Server, SignalR, Web Sockets, Onion Architecture

- Integrated Redis as a message buffer between WebSocket streams and the application layer, ensuring zero data loss under highthroughput real time market data ingestion.
- Designed domain models for transaction history, user balances, and portfolio tracking, ensuring data consistency through EF Core transaction management.
- Implemented Clean Architecture with repository pattern, enabling independent testability across all service layers.

Clean Bank [🔗](#) [🔗](#) | Java 21, Spring Boot 4, Next.js 16 PostgreSQL, RabbitMQ, Domain Driven Design, CQRS

- Architected a domain-driven core banking platform across four Maven modules enforcing Onion Architecture with a CQRS and MediatR
- Implemented Transactional Outbox Pattern with RabbitMQ consumers for reliable async event processing and applied locking with version-based conflict detection to prevent concurrent balance corruption.
- Containerized the full stack with Docker Compose and deployed to AWS EC2 behind Caddy reverse proxy with automatic Let's Encrypt TLS provisioning.
- Built a REST API, pagination with statistics aggregation, and JWT authentication with a Next.js 16 frontend featuring a reusable component library and Axios interceptors for token lifecycle management.

Education

Eskisehir Technical University

Expected June 2026

Bachelor of Science in Computer Engineering

Eskisehir

- Relevant Coursework:** Data Structures and Algorithms, Object-Oriented-Programming, Prob & Stat in CS, Linear Algebra w/Computational Applications, Computer Vision, Machine Learning

Technical Skills

Programming Languages: Java, C#, TypeScript, JavaScript, SQL

Backend: Spring Boot, .NET Core, JPA/Hibernate, Entity Framework Core, REST API

Frontend: Next.js (App Router), React, Angular, Tailwind CSS

Data & Messaging: PostgreSQL, MySQL, Redis, Apache Kafka, RabbitMQ, SignalR

Architecture: Microservices, DDD, CQRS, Clean/Onion Architecture, Outbox-Saga Pattern, Event-Driven Architecture

DevOps & Tools: Docker, Git, Prometheus, Grafana, Seq

Languages

Turkish: Native

English: Professional Working Proficiency